Spring 2023

Human Tracking Function for Robotic Dog

Andrew Sharkey
acs169@uakron.edu

Human Tracking Function for Robotic Dog

SENIOR DESIGN

MECE 471

HONORS PROJECT

MECE 497

By

Andrew, Sharkey
UA Advisor: Dr. Yalin Dong

Dept. of Mechanical Engineering
College of Engineering
University of Akron

April 2023

Abstract

With the increase the increase in automation and humans and robots working side by side, there is a need for a more organic way of controlling robots. The goal of this project is to create a control system for Boston dynamics robotic dog Spot that implements human tracking image software to follow humans using computer vision as well as using hand tracking image software to allow for control input through hand gestures.

Contents

Contents

1. Introduction

1.1 Project Overview

With the increase in utilization of robotics in industry and everyday life, it is necessary to implement a way for robots to interact and perform task alongside humans in a safe and effective manner. The integration of robots and automation in both industry and everyday life is steadily growing more and more common. With this increase comes the need to develop ways of controlling robots that are more user friendly. Boston Dynamics' robots are especially suited to this application because the care and thought that Boston Dynamics puts into making their robot safe for those around them. In their statement of ethical principles, they say: "It is imperative that robots be trustworthy if people are to work productively with them. Therefore, we will build robots whose missions and capabilities are predictable, understandable, transparent and in service of human needs" (Boston Dynamics Ethical Principles). In order for robots to work effectively alongside humans, powerful motors and actuators must be used to fulfill motion requirements of robots such as Spot. Thus, extensive safety measures must be implemented to avoid injuries. In this project, safety was our primary concern, and we first considered the safety of those around the robot in our design parameters.

In addition to this paper, there is a video of a demonstration of the robot and other codes that are more easily shown than explained. The link is as follows: https://youtu.be/Cb5cgDhNxdo.

1.2 Goals and Objectives

Our project's goal was to design, deploy, and test human tracking for the Boston Dynamics' robotic dog Spot and use alternative inputs such as hand motions to control the robot. Several human tracking software already exist so all we needed to do is apply the existing solutions to the robot. The requirements for the design are as follows.

The implemented system must:

- Have human and hand tracking capabilities.
- Respond to minimum three hand motion commands.
- Be able to follow human at specified distance.
- Have built in safety protocols to mitigate danger.

1.3 Background information

Google's Mediapipe framework comprises a pre-built machine learning library, which is intended for use in real-time applications. The algorithms encompass a range of functions, from detecting objects and faces, to tracking hands and estimating poses. In our project, we used Mediapipe's holistic module which combines the pose, face, and hand modules to create

landmarks for the entire human body.

Boston Dynamics has a Python API that allows programmers to communicate with Spot robot using Python. The Spot Python API provides a variety of functions that enable users to perform tasks such as controlling the robot's movements and collecting data from its onboard sensors. We utilized this functionality of Spot to direct commands to Spot through our own python code.

1.4 Engineering standards applicable to the project

Regarding standards, ISO/TS 15066 talks about how to safely implement systems that incorporate collaborative interactions between robots and humans (OSHA). In comparison to industrial robots who work safely in their own space, the safety requirements for robots who share space with human workers are much stricter. This ISO requirement outlines many thresholds such as force, speed and pressure exerted by the robot to avoid injury to humans.

2. Implementation process

The project was split into two separate phases. During phase 1, we explored and tested the capabilities of the software development kit (SDK) supplied by Boston Dynamics as well as MediaPipe's machine learning solutions including the hand module for hand detection and the pose module for human detection. By the end of Phase 1, we had a working program using both software kits to test the feasibility of the project.

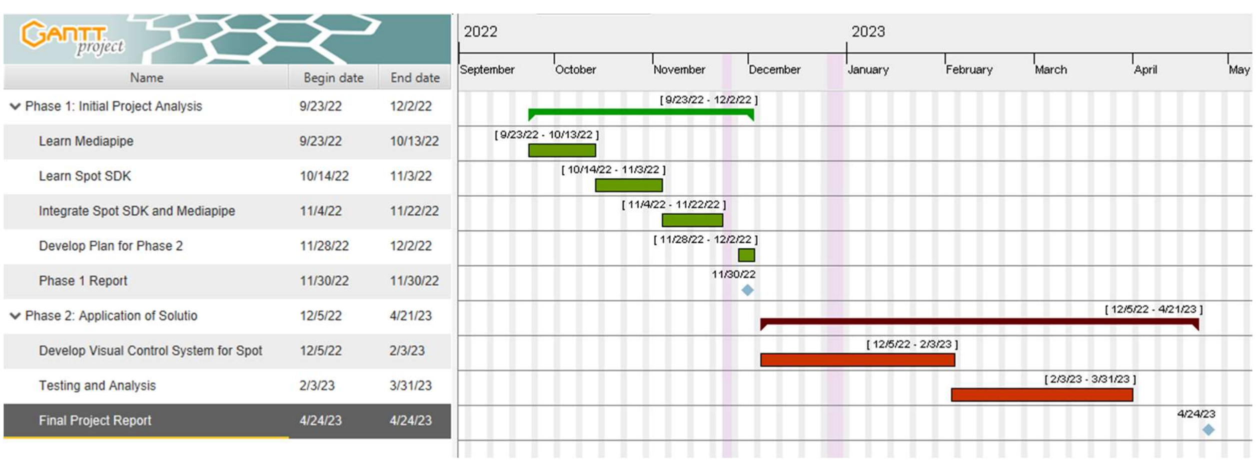


Figure 1: Project Schedule

During Phase 2 of the project, we developed the control system for the human following program. After designing the control system, we developed the Python code that implemented the control system and integrated it with the MediaPipe image processing code and the command development code for control over Spot. After successfully implementing the code on the robot, we also explored ways of improving the performance of the system and investigated other hardware systems for improved performance and functionality.

3. Design and Methodology

3.1 Project design and methodology

The system hierarchy is shown in the diagram below. The system will contain three separate hardware systems: the camera, the robotic dog Spot, and the computer.

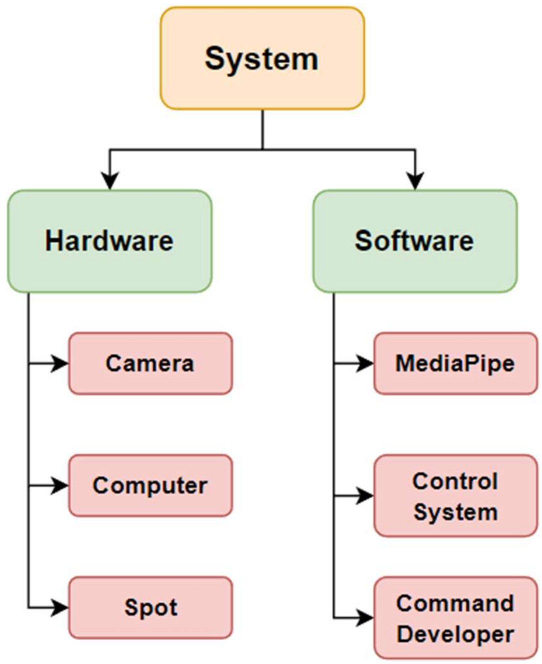


Figure 2: System Hierarchy

The camera will collect information about the robot's surroundings and send the videos to the computer where the computer will analyze the images using the MediaPipe software. After the images are analyzed, the data will be input into the control system software which will determine the desired response of the robot. The command developer subroutine will then create the command that can be interpreted by the robot and the computer will send the command to the robot through the Wi-Fi connection. In addition to sending the command the computer will output the processed video onto the computer display for the user.

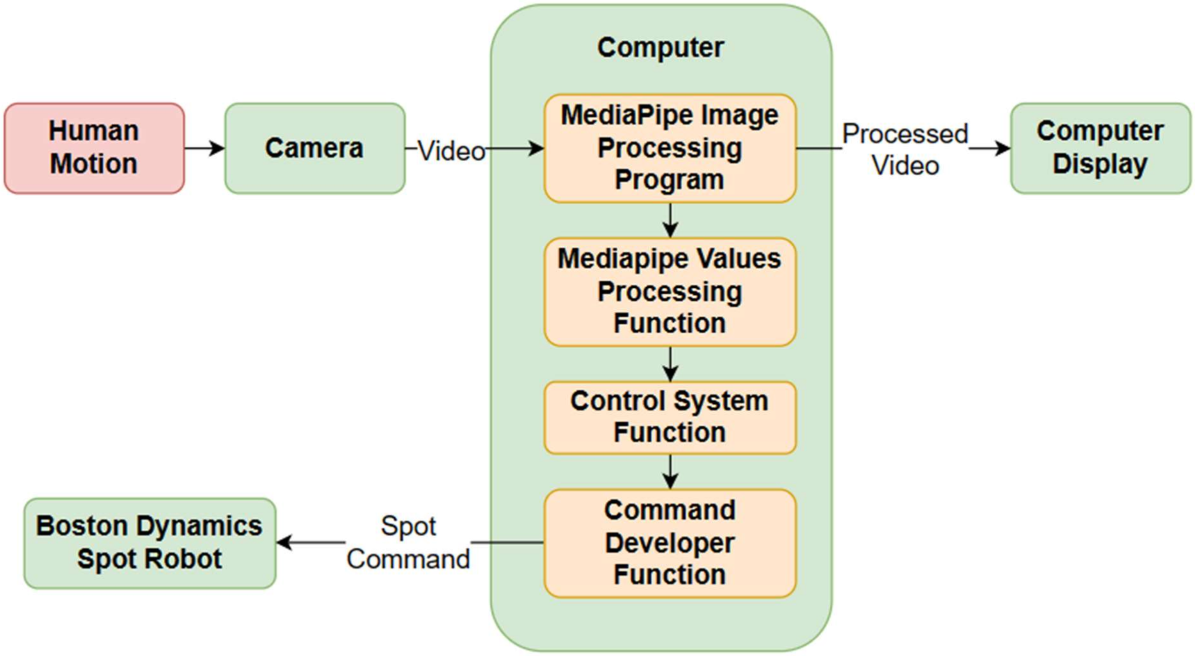


Figure 3: Signal Flow Diagram

Our main program consisted of several different parts, each performing many different tasks. The beginning of the code consists of several different initializations. Before commands can be sent to Spot, an initialization process must first take place, which includes grabbing a lease from the robot. This lease is what tells the robot who can issue commands. Once someone has grabbed a lease no one else is able to grab the lease until it is returned. In addition to Spot initialization, the Mediapipe model must be initialized with certain parameters.

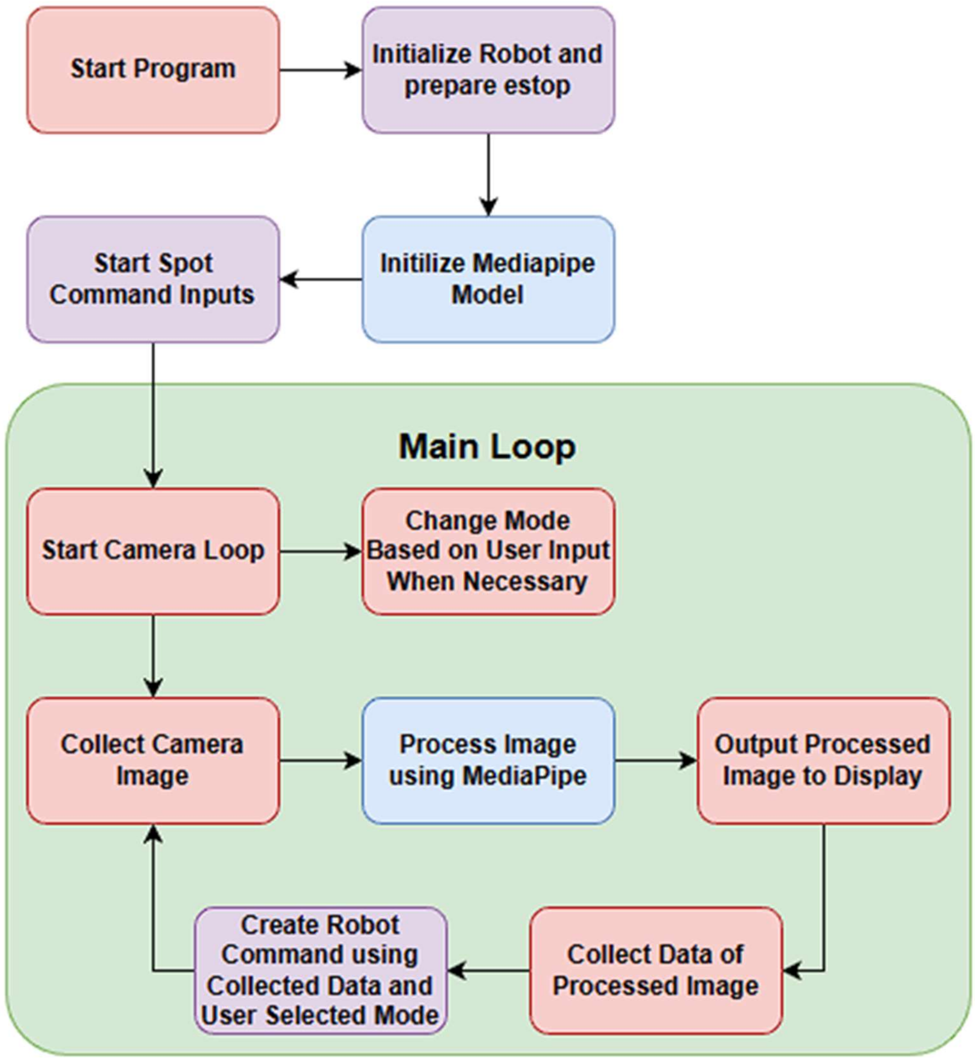


Figure 4: Code Structure

After all of the initialization code, we first direct Spot to stand up, then begin the main loop. During each loop, the code will collect the camera image and process it using the MediaPipe Model. Then it will output the processed image to the computer display and create a robot command using the landmark data and the user selected mode. During all times during the loop, the code listens for a user input through keystrokes. If the user presses any of the keys in the following table, the code will adapt to follow the instructions.

Input	Action
1	Mode 1
2	Mode 2
3	Mode 3
4	Mode 4
S	Exit loop
Spacebar	Pause loop

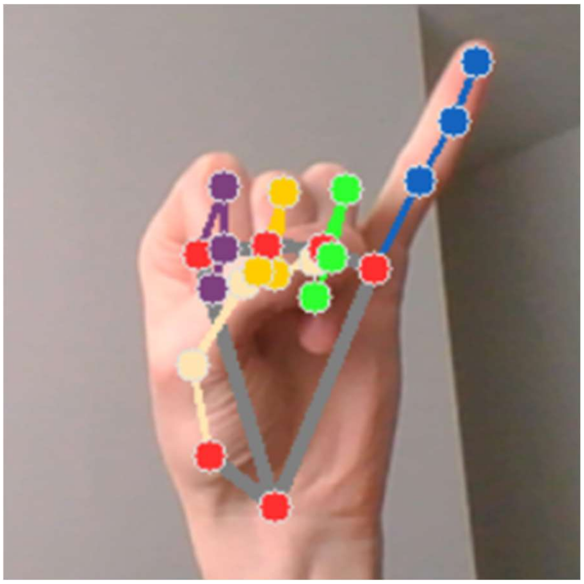
Table 1: User input options

To allow for extra options we created the several different modes to control what types of motions the robot will perform. When the program first begins, mode 4 is selected. While in mode 4, no commands will be sent to the robot, however the loop will continue and the MediaPipe processing will still occur. In mode 1, commands to the robot will only change the stance of the robot, such as pitch and yaw, to turn to follow the person's hand who is in the camera frame. There is a limit to how far the robot will contort to prevent Spot from losing its balance, thus this mode can only be utilized for people standing directly in front. In mode 2, motion commands that changes the robots rotation are implemented as well, allowing for a 360 degree range of motion. In mode 3, the code will also send motion commands that move Spot forward and backwards in addition to its pitch and rotation.

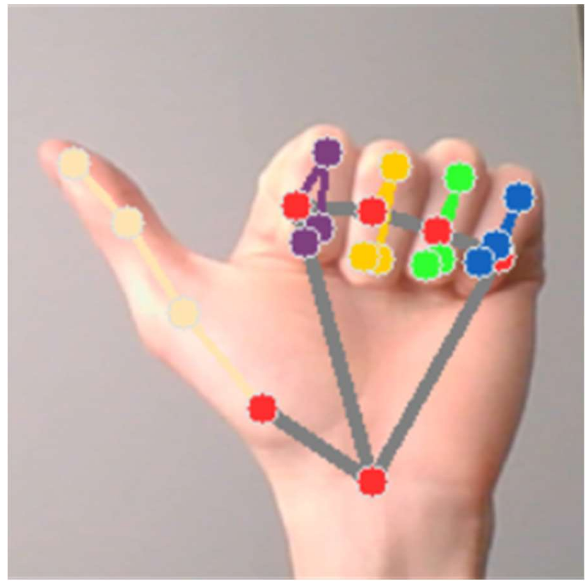
Mode	Short Description
1	Change pitch and yaw to follow hand in camera frame
2	Change pitch and rotate in place to follow hand in camera frame
3	Change pitch and rotate to follow hand in camera frame as well as walk forwards or backwards to follow human
4	No robot motion

Table 2: Mode options

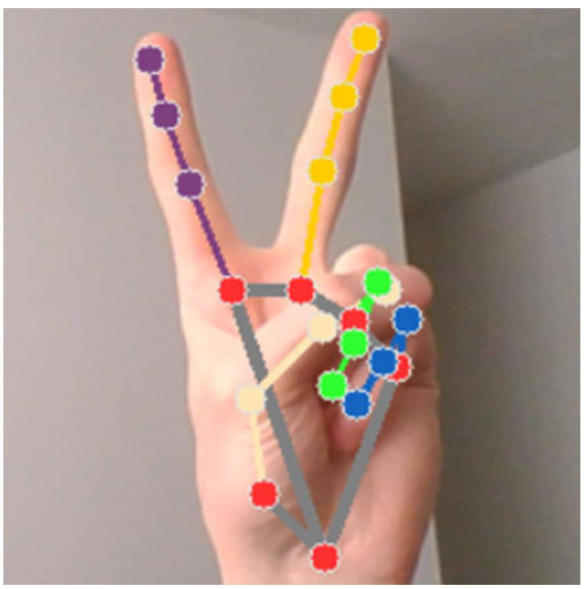
For the first three modes, in addition to their hand following motions, the robot will also respond to different hand commands. The four hand commands we implemented are shown on the table below. When the program sees the fingers held up like what is shown in the photo the robot will perform a command as described in the caption. For example, to make the robot tilt to the right, the user can hold out their thumb.



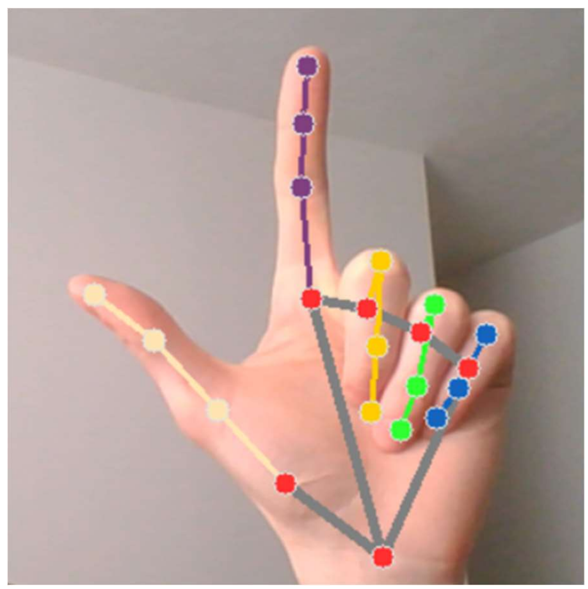
Tilt Head Left



Tilt Head Right



Stand Taller



Stand Shorter

Figure 5: Various Hand Commands

3.2 Technical approaches and decisions

We investigated several solutions to different hardware systems to improve our initial design. In addition to a basic laptop, we looked into using a Raspberry Pi and the Spot Core IO Payload so we could run the software on the robot without wires. The Core IO Payload is a computer produced by Boston Dynamics to attach to the back of Spot and run using Spot's battery. It would have

processing power comparable to a laptop, thus allowing for the same reaction speed, while eliminating the need to have a wire, allowing for more maneuverability for Spot. However, we were never able to get the payload to access the camera while attached to Spot, thus we were unable to use it. Regarding the Raspberry Pi, there were many issues. When installing the necessary Python libraries on the Pi, we found that the Mediapipe library we had been using, did not work. Thankfully we were able to circumvent this issue by using a version of Mediapipe specially created for use with the Raspberry Pi. However, even with this version of Mediapipe, the processing speed of the video was too slow for our system. Below is the trade study we used to analyze help us decide which system to go with.

Criteria	**Weight**	**Laptop Computer**		**Raspberry Pi**		**Spot Core IO Payload**	
Computational Speed	0.3	10	3	4	1.2	10	3
Wireless	0.2	1	0.2	8	1.6	10	2
Software Compatibility	0.3	10	3	3	0.9	8	2.4
Power Source	0.2	10	2	5	1	8	1.6
	1		**8.2**		**4.7**		**9**

Table 3: Computer Trade Study

4. Conclusion

In addition to hand commands, integrating voice controls with Boston Dynamics' Spot robot could also offer a more natural and intuitive way to interact with the robot. By leveraging advances in speech recognition and natural language processing, users could issue voice commands to direct the robot's movements, manipulate objects, or perform various tasks. Such integration could also enhance accessibility and ease of use for individuals with disabilities, who may have difficulty using traditional input devices like keyboards or joysticks. Moreover, with the increasing capabilities of AI language models such as ChatGPT and Google's Bard, a program could be developed for Spot such that the robot would be able to converse and interact with those around it using text to speech.

In conclusion, with the increasing use of robotics in industry and everyday life, it is necessary to develop ways for robots to interact and perform tasks alongside humans in a safe and effective manner. The integration of robots and automation in both industry and everyday life is becoming more and more common, and the need for more user-friendly ways of controlling robots is growing. Overall, this project demonstrated the feasibility of using machine learning and robotics to track and follow humans for collaborative interactions between robots and humans.

5. References

Boston Dynamics. (n.d.). *Spot SDK 3.0.3 Documentation* Dev.bostondynamics.com. Retrieved November 26, 2022, from https://dev.bostondynamics.com/

Google LLC. (n.d.). *MediaPipe Hands* google.github.io. Retrieved November 26, 2022, from https://google.github.io/mediapipe/solutions/hands.html#:~:text=MediaPipe%20Hands%20is%20a%20high-fidelity%20hand%20and%20finger,of%20a%20hand%20from%20just%20a%20single%20frame.

Boston Dynamics. (n.d.). *Boston Dynamics Ethical Principles* www.bostondynamics.com. Retrieved November 26, 2022, from https://www.bostondynamics.com/ethics

OSHA. (n.d.). *Robotics - Standards | Occupational Safety and Health Administration*. (n.d.). www.osha.gov. Retrieved November 29, 2022, from https://www.osha.gov/robotics/standards#:~:text=1%20TC%20299%2C%20Robotics.%20...%202%20ISO%2010218-1%2C

Tangible Engineering. (2023, April 13). Senior Design Project: Equip Boston Dynamics Spot with AI vision [Video]. YouTube. https://youtu.be/Cb5cgDhNxdo